nocoRNAc User Guide
v 1.21

Alexander Herbig
alexander.herbig@uni-tuebingen.de

Kay Nieselt
kay.nieselt@uni-tuebingen.de

February 1, 2011

1 General Instructions

nocoRNAc (non-coding RNA characterization) is a Java program for the prediction and characterization of ncRNA transcripts in bacteria. nocoRNAc takes the coordinates of putative ncRNA loci as input and annotates them with transcriptional features to conduct strand-specic transcript predictions. Our approach is not limited to intergenic regions but also applied to predict cis-encoded asRNA transcripts.

The directory of the nocoRNAc distribution contains the `nocoRNAc.jar` file. The configuration file `config.conf`, as well as the directories `data` and `progs`. The directories for the result files for different genomes will be generated in the `data` directory. nocoRNAc needs Java version 1.6 or higher. Some basic examples for nocoRNAc's application are provided in the following section. A description of all command line options and all configuration file entries is given in sections 3 and 4. A list of programs that can be utilized by nocoRNAc can be found in section 6.

2 Standard Protocols

The following two command lines are used for a complete nocoRNAc run using all standard procedures.

SIDD profile The following command line is used to generate the SIDD profile for a genome. Depending on the size and sequence composition of the genome this may take several days. To speed up the calculation adjust the **DEFAULT_BIG_WINDOW_SIZE** and **DEAFAULT_BIG_WINDOW_SHIFT** parameters in the configuration file.

```
java -Xmx1G -jar nocoRNAc.jar -genomeFastaFile <filename> -ncRNAcrdfile
<filename> -siddProfile
```

GFF output The following command line is used to perform the less time consuming procedures (i.e. **-siddSites**, **-terminators**) and to generate the GFF output:

```
java -Xmx1G -jar nocoRNAc.jar -genomeFastaFile <filename> -ncRNAcrdfile
<filename> -proteinTableFile <filename> -siddSites -terminators
-gffOutFile <filename>
```

Note that **-terminators** can be replaced by **-importTransTerm FILE** to import Rho-independent terminator annotations in TransTermHP format. Such annotations can be retrieved for many bacterial genomes from the TransTermHP website (`http://transterm.cbcb.umd.edu/`).

In general, it is also possible to put all procedures in one command line, but it is recommended to separate the more time consuming procedures from each other. In fact, all procedures can be performed during separate runs, as the results are stored in the directory of the given genome. These results are read when the **-gffOutFile** or **-count** arguments are used.

Command line examples for the more specialized methods of nocoRNAc are listed below. A description of all command line options and all configuration file entries is given in the following two sections.

Rfam scan The following command line is used to search for Rfam motifs within the predicted ncRNA regions. The Rfam motifs that are searched have to be specified in the file passed with the **-rFamScan** option. The proceure will take several hours to several days depending on the number of motifs that are searched. Use the **-rFamAll** option instead to search for all motifs

contained in the Rfam seed file. If a search for all motifs is done, the used of a mutli processor system is highly recommended. Use the **-numCPUs** option to specify the number of parallel processes. *Erpin* and *infernal* have to be installed correctly (see section 6).

```
java -Xmx1G -jar nocoRNAc.jar -genomeFastaFile <filename> -ncRNAcrdfile
<filename> -rFamScan <filename> -rFamSeeds <filename>
```

RNA-RNA interactions The following command line is used to predict RNA-RNA interactions between ncRNAs (**-hunters**) and the mRNAs of protein coding genes (**-targets**). Depending on the number of elements and the length of their sequence this might take several days. If the procedure is applied to hundreds of ncRNAs and coding genes it probably takes several weeks. The program *IntaRNA* has to be properly installed (see section 6).

```
java -Xmx1G -jar nocoRNAc.jar -genomeFastaFile <filename> -ncRNAcrdfile
<filename> -proteinTableFile <filename> -interactions -hunters <filename>
-targets <filename>
```

Interaction profiles/matrix The following command line is used to calculate interaction profiles and an interaction matrix on the basis of previously predicted RNA-RNA interactions.

```
java -Xmx1G -jar nocoRNAc.jar -genomeFastaFile <filename> -ncRNAcrdfile
<filename> -proteinTableFile <filename> -interactionProfiles
-interactionMatrix
```

Result plots The following command line generates a PDF file in the genome directory containing plots of predicted ncRNA regions in the context of the features which are detected by nocoRNAc. The file passed with this argument has to contain the IDs of the ncRNA regions for which plots shall be generated. To produce single plots in JPG format use **-plotJPGncRNARegions**.

```
java -Xmx1G -jar nocoRNAc.jar -genomeFastaFile <filename> -ncRNAcrdfile
<filename> -proteinTableFile <filename> -plotPDFncRNARegions <filename>
```

R environment Use the following commandline to open the interactive `R` shell, which provides large parts of nocoRNAc's data structure for the purpose of statistical analysis and visualization using the programming language `R`.

```
java -Xmx1G -jar nocoRNAc.jar -genomeFastaFile <filename> -ncRNAcrdfile
<filename> -proteinTableFile <filename> -rFamSeeds <filename> -siddSites
-terminators -pcMatches <filename> -count
```

3 Command line options

The interpretation of commandline options is case insensitive. So instead of writing '-genomeFastaFile' the user can also use '-genomefastafile'. In addition, an option can be abbreviated using a non-redundant prefix, e.g. '-genome' instead of '-genomefastafile'.

-projectname NAME By default the name of the subfolder in the data folder that contains the result files for a specific genome is set automatically by nocoRNAc using the genome name from the genome FASTA file. The **-projectname** option can be used to set the name manually. Note that after the name has been manually set this option has to be used each time nocoRNAc is started on the respective genome. Otherwise previously calculated results will not be found.

-genomeFastaFile FILE This argument is used to provide a FASTA file containing the genomic sequence to which nocoRNAc is applied. If a multiple FASTA file is given, only the first entry will be read. The name of the sequence, which is given in the ID line (> ...), will be used to determine the name of the directory in which the results for this genome are stored. This is a mandatory argument.

-ncRNAcrdFile FILE This argument is used to provide a file containing the IDs and genomic coordinates of predicted ncRNA regions. The file can be in one of the following 2 formats:
1. A simple white-space separated format, where each line contains the following 4 fields:
ID start end strand
strand can be '+', '-' or '.'. The fields must not contain white-space characters. The content should be sorted with respect to the start positions. nocoRNAc will run slower if the entries are not sorted.

2. A file in GFF format. The IDs of the regions have to be provided by an attribute named 'ID'. If no IDs are contained in the GFF file, they are generated by NOCORNAC. The GFF output of the RNAz script `rnazIndex.pl` can directly be used as NOCORNAC's input. This is a mandatory argument.

-proteinTableFile FILE This argument is used to provide a file containing protein coding regions. It has to be in Protein-Table format (`.ptt`). These files can be retrieved from the NCBI ftp server. For example:
`ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Streptomyces_coelicolor/NC_003888.ptt`

-gffOutFile FILE This argument is used to generate a file in GFF format containing all results of previously applied methods. It contains predicted ncRNA regions (including classification results), protein coding regions (if provided), terminator signals, SIDD sites, sigma factor binding sites, Rfam motif matches and predicted ncRNA transcripts. Only features for which a respective result file is found are considered. Rfam motif matches can only be properly contained if an Rfam seed file is provided using the **-rFamSeeds** argument.

-siddProfile If set, the SIDD profile for the given genome will be calculated. This will take several hours! The profile will be stored in the file `siddProfile.out` in the directory of the given genome.

-siddSites If set, the SIDD profile of the given genome is analyzed to detect SIDD sites. The detected SIDD sites will be stored in the file `siddSites.out` in the directory of the given genome. SIDD sites can only be detected, if the SIDD profile for that genome has been calculated.

-terminators If set, Rho-independent terminator signals are detected in the given genome using the program TransTermHP. The results will be stored in the file `terminators.out` in the directory of the given genome. Terminator detection usually takes less than a minute.

-importTransTerm FILE If set, Rho-independent terminator annotations in TransTermHP format will be imported for the given genome from the provided file. Such annotations can be retrieved for many bacterial genomes from the TransTermHP website (`http://transterm.cbcb.umd.edu/`). The annotations will be stored in the file `terminators.out` in the directory of

the given genome. This command can be used instead of the **-terminators** command.

-pcMatches FILE If set, transcription factor binding sites are searched in the given genome. The results will be stored in the file `pcMatches.out` in the directory of the given genome. A file containing sequence patterns in the form of regular expressions, which represent the binding sites, has to be provided. The regular expressions have to fulfill the Java conventions for regular expressions. Each line of the file has to be of the form:
name of the binding site regular expression
Example line:
factor1 TCAGTC[AT]{16}TCGA
The two fields have to be separated by white-space and must not contain white-space characters.

-rFamSeeds FILE This argument is used to provide Rfam seeds which represent RNA motifs which are contained in the Rfam database.
(`ftp://ftp.sanger.ac.uk/pub/databases/Rfam/CURRENT/Rfam.seed.gz`)
This argument is mandatory, if **-rFamScan** or **-rFamAll** are set.

-rFamScan FILE If set, all predicted ncRNA regions will be scanned for motifs contained in the Rfam database. The motifs have to be provided as seeds using the **-rFamSeeds** argument. A file has to be provided (FILE) which contains keywords related to the motifs which shall be searched. The keywords can be Rfam accession numbers (e.g. RF00001) or Rfam entry types. The scan will be performed for all seeds whose accession number or entry type matches one of the keywords. For a description of Rfam entry types see:
`ftp://ftp.sanger.ac.uk/pub/databases/Rfam/CURRENT/USERMAN`
The keywords have to be separated by white-space or newline and must not contain white-space characters. The results are stored in the file `rfamhits.out` in the directory of the respective genome.

-rFamAll Like **-rFamScan**, but no file containing keywords has to be provided. The scan is performed for all motifs which are contained in the provided seed file (**-rFamSeeds**). This will take several hours!

-appendRfamHits If set, the motifs which are detected when one of the above arguments is set are appended to the file **rfamhits.out** if it already exists. To overwrite the existing file the **-overwrite** option has to be set.

-compare2Annotation FILE This option is used to pass a GFF file containing annotations for the current genome to which the ncRNA regions and predicted ncRNA transcripts will be compared. This can be, for example, annotations of known ncRNAs in that genome (e.g. from Rfam). The results are written to stdout and contain a mapping of ncRNA regions and predicted transcripts to the provided annotations, as well as some statistics like the number of annotations overlapping a predicted ncRNA region or transcript etc.

-overwrite All procedures mentioned above will not be performed if the respective result file already exists, as overwriting is disabled by default. If the -overwrite option is set, all applied procedures will overwrite already existing result files.

-plotPDFncRNARegions FILE If this argument is used, a PDF file is generated in the genome directory containing plots of predicted ncRNA regions in the context of the features which are detected by NOCORNAC. The plots show ncRNA regions, protein coding genes, predicted terminator signals and the SIDD profile of the respective region.
The file passed with this argument has to contain the IDs of the ncRNA regions for which plots shall be generated. The IDs have to be separated by white-space or newline.
Instead of a file path the keyword `all` can be used. In this case for all ncRNA loci a plot is generated. This is not recommended if there are more than about 100 loci.
It is also possible to pass a comma separated list of IDs instead of a file path. When doing this the list must not contain any white space.
Only features for which a respective result file is found are considered. At least the file `siddProfile.out` has to exist. NOCORNAC uses `R` to generate the plots. Therefore `R` has to be installed on the system.

-plotJPGncRNARegions FILE Like **-plotPDFncRNARegions**, but instead of a single PDF file several JPG files are generated, which contain the plots.

-interactions If set, RNA-RNA interactions are predicted using the program IntaRNA. The interactions are predicted between a set of what we call 'hunters' (usually a set of predicted ncRNA regions) and a set of, so called, 'targets' (usually a set of protein coding genes). The resulting predictions are

stored in the file `interactions.out`, which will be located in the genome directory. Each hunter or target has to be contained either in the set of ncRNA regions (**-ncRNAcrdFile**) or protein coding genes (**-proteinTableFile**). The IDs of hunters and targets have to be provided by the use of the arguments **-hunters** and **-targets**.

-hunters FILE Use this argument to provide a file containing IDs of ncRNA regions for which RNA-RNA interactions with mRNAs of protein coding genes shall be predicted (see **-interactions**). As ncRNA regions are potentially not strand specific NOCORNAC will apply the interaction prediction to both strands of an ncRNA region, if its strand is not specified in the coordinates file (**-ncRNAcrdFile**).

-targets FILE Use this argument to provide a file containing IDs of protein coding genes for which RNA-RNA interactions with predicted ncRNA regions will be predicted (see **-interactions**).

-appendInteractions If set, the interactions which are predicted when **-interactions** is set are appended to the file **interactions.out** if it already exists. To overwrite the existing file the **-overwrite** option has to be set.

-importIntaRNA FILE If RNA-RNA interactions have been predicted with IntaRNA without using NOCORNAC, this argument can be used to import an IntaRNA output file and to create a NOCORNAC compatible file (`interactions.out`) in the genome directory. If `interactions.out` already exists, the **-appendInteractions** option can be used to append the imported interactions. Each interaction partner in the IntaRNA output has to be contained either in the set of ncRNA regions (**-ncRNAcrdFile**) or protein coding genes (**-proteinTableFile**). Otherwise the respective interaction is discarded.

-DOTfromInteractions If set a file in DOT format (`interactions.dot`) will be generated in the genome directory containing the description of a graph representing the RNA-RNA interaction network in `interactions.out`. If this argument is not combined with **-interactions** the file `interactions.out` has to exist already.

-GMLfromInteractions If set a file in GML format (`interactions.gml`) will be generated in the genome directory containing the description of a graph representing the RNA-RNA interaction network in `interactions.out`.

If this argument is not combined with **-interactions** the file `interactions.out` has to exist already.

-interactionProfiles If set files will be generated in the genome folder containing calculated interaction profiles for each element of previously predicted interactions (`iProfileN.tsv`, `iProfileG.tsv`, `iProfileP.tsv`, `iProfilePnet.tsv`) We refer to the paper for a detailed description of the profiles. If this argument is not combined with **-interactions** the file `interactions.out` has to exist already.

-interactionMatrix If set a file will be generated in the genome folder (`iMatrix.tsv`) containing the calculated interaction matrix for previously predicted interactions. If this argument is not combined with **-interactions** the file `interactions.out` has to exist already.

-numCPUs N The number of parallel processes that will be started by noCoRNAc (default: 1). Several functions of noCoRNAc can be run in parallel. This includes the calculation of the SIDD profile, the scan for RNA structure motifes and, partially, the RNA-RNA interaction prediction. Even if a multiprocessor system is available, the user might not want to use all precessors. Therefore the number of parallel processes can be limited by the use of this argument.

-help If set, a list of all possible command line arguments and a short description of each argument is written to stdout.

-version Prints the version number and built date of noCoRNAc.

-count If set, noCoRNAc provides parts of its data structure within an interactive `R` shell allowing the user to perform a variety of statistical analysis on the results as well as visualizing them by the use of all functionalities of the programming language `R`. In principle the results of almost all functions of noCoRNAc will be accessible, if the respective result file is found. This includes the genome sequence, the list of protein coding genes, a nested list structure of all predicted ncRNA regions with all annotated features, predicted terminator signals, the SIDD profile, SIDD sites, detected sigma factor binding sites, predicted ncRNA transcripts, RNA-RNA interactions, interaction profiles and the interaction matrix.

4 Configuration File

The file `config.conf` which is contained in the nocoRNA directory contains several settings influencing nocoRNAcs application. The important settings are described in this section. It is not recommended to change values which are not described here. If nocoRNAc cannot find a configuration file at its location, it creates one with default values for all settings.

commentChar (default '#') If a line in a file which is provided to nocoRNAc starts with commentChar, this line is not read.

dataPath (default 'data') Name of the directory in which result folders will be created for each genome. If the given path is preceeded by an '/' it is interpreted as an absolute path (e.g. /home/user/myfolder). Otherwise it is interpreted relative to the actual path. This is also true for all options concerning paths.

transtermPath (default 'progs/transterm') Path to the program files of TransTermHP. The respective folder has to contain the executable binary (`transterm`) and the file `expterm.dat` (part of the TransTermHP distribution).

transtermConfidenceCutoff (default 50) Cutoff for the confidence value of terminator signals. Only terminators with a confidence value above the cutoff will be considered in the various procedures.

siddEnergyCutoff (default 6.0) Cutoff for the incremental energy value of SIDD sites. Only SIDD sites with a energy value below the cutoff will be considered in the various procedures.

termDownstreamRegionSize (default 25) Size of the downstream region of genes or the flanking regions of predicted ncRNA regions in which a terminator signal has to be located to be considered to potentially belong to the gene or ncRNA region respectively.

pcUpstreamRegionSize (default 25) Size of the upstream region of genes or the flanking regions of predicted ncRNA regions in which a predicted transcription factor binding site has to be located to be considered to potentially belong to the gene or ncRNA region respectively.

siddUpstreamRegionSize (default 25) Size of the upstream region of genes or the flanking regions of predicted ncRNA regions in which a predicted SIDD site has to be located to be considered to potentially belong to the gene or ncRNA region respectively.

siddRelativeStart (default 0.0) Value between 0.0 and 1.0. The start of predicted ncRNA transcripts is by default set to the start position of the assigned SIDD site (0.0). This ensures a high probability for the actual transcript to be completely contained in the predicted region. However, the average deviation between predicted start positions and actual start positions is minimal if the predicted start is set to a position located in the second half of the SIDD site (e.g. 0.8).

erpinPath (default 'progs/erpin') Path to the program files of Erpin. The binaries have to be located in the `bin` subfolder. The Perl scripts, including `erpincommand.pl`, have to be located in the `scripts` subfolder.

erpinSourceName (default 'erpin') This String is used to indicate in the GFF output that a detected Rfam motif has been found by Erpin.

infernalPath (default 'progs/infernal') Path to the program files of CMsearch. The specified folder has to contain the two executable binaries `cmbuild` and `cmsearch`.

cmsearchEValueCutoff (default 0.01) The E value cutoff used during the application of CMsearch.

cmsearchSourceName (default 'cmsearch') This String is used to indicate in the GFF output that a detected Rfam motif has been found by CMsearch.

cmsearchOnly (boolean: 0 or 1, default 0) If set to 1, Erpin will not be used when scanning the ncRNA regions for RNA motifs.

allowNonTerminatorTranscripts (boolean: 0 or 1, default 0) If set to 1, the prediction of ncRNA transcripts without a terminator signal is allowed.

noTerminatorSelection (boolean: 0 or 1, default 0) If set to 1, there will be a predicted ncRNA transcript for each terminator signal not only for the best one. Note that this might result in overlapping transcripts unless **mergeOverlappingTranscripts** is set to 1.

mergeOverlappingTranscripts (boolean: 0 or 1, default 1) If set to 1, predicted ncRNA transcripts that overlap and that are located on the same strand are merged, if they belong to the same ncRNA locus.

forceStrandSpecification (boolean: 0 or 1, default 1) If set to 1, predicted ncRNA transcripts that are located on different strands must not overlap. In such a case the transcript with the stronger signals (terminator, sidd site) will occupy the respective region. The other transcript is shortened or completely discarded.

transcriptPredictionIgnoreStrand (boolean: 0 or 1, default 0) If set to 1, the prediction of ncRNA transcripts is applied to both strands of a predicted ncRNA locus even if the locus has a strand specified. Otherwise the procedure is only applied to the specified strand. If the strand of a ncRNA locus is unknown, the procedure is also applied to both strands.

plotFlankingRegionSize (default 200) Size of the flanking regions of predicted ncRNA transcripts for which plots are generated using the **-plotPDFncRNARegions** or **-plotJPGncRNARegions** arguments. The flanking regions will also be shown in the plot.

intarnaPath (default 'progs/intarna') Path to the program files of IntaRNA. The specified folder has to contain a subfolder (`bin`) containing the executable binary `IntaRNA` and a `lib` subfolder containing the file `libRNA.a`. This library can be taken from the Vienna package. It is generally essential when using IntaRNA.

intarnaLengthSwitch (default 150) For longer ncRNAs or mRNAs a sliding window is used.

intarnaAdjustLengthSwitch (boolean: 0 or 1, default 1) If set to '1' and if the shortest target is shorter than **intarnaLengthSwitch**, then **intarnaLengthSwitch** will be set to the length of the shortest target. If set to '0', targets shorter than **intarnaLengthSwitch** are not processed.

intarnaSeedLength (default 8) Minimal length of the seed region. For details we refer to the manpage of IntaRNA.

intarnaSeedMM (default 1) Maximal number of allowed missmatches in the seed region. For details we refer to the manpage of IntaRNA.

intarnaPreserveOutput (boolean: 0 or 1, default 1) noCoRNAc stores IntaRNAs output in the genome folder of the current genome. If this option is set to '0', the output is deleted after processed by noCoRNAc. If set to '1', the original IntaRNA output is preserved. However it will be overwritten during subsequent interaction predictions for the same genome.

autoThres (boolean: 0 or 1, default 1) When generating DOT or GML graph files from interaction networks (**-DOTfromInteractions** or **-DOTfromInteractions**) the interactions are filtered to reduce the number of edges. If this option is set to '1' the threshold used during this filtering is determined automatically by noCoRNAc by only considering interactions whose free energy value and/or size of the interacting region is better than a specified percentile of the respective distribution. These percentiles can be specified using the options described in the following.
If set to '0' fixed thresholds are used, which can also be specified (**veryGoodEnergyThres**, etc.).

veryGoodEnergyPerc (default 1.0) Percentile-threshold for very good interaction energy.

veryGoodLengthPerc (default 1.0) Percentile-threshold for very good interaction length.

goodEnergyPerc (default 2.0) Percentile-threshold for good interaction energy.

goodLengthPerc (default 2.0) Percentile-threshold for good interaction length.

veryGoodEnergyThres (default -25.0) Threshold for very good interaction energy.

veryGoodLengthThres (default 50) Threshold for very good interaction length.

goodEnergyThres (default -20) Threshold for good interaction energy.

goodLengthThres (default 10) Threshold for good interaction length.

veryGoodEnergyCol (default '#C6241A' (red)) Color of very good interaction energy edges.

veryGoodLengthCol (default '#1410D0' (blue)) Color of very good interaction length edges.

goodEnergyAndLengthCol (default '#8B08B9' (purple)) Color of good interaction energy and length edges.

senseAntisenseCol (default '#000000' (black)) Color of sense antisense pairs.

DEFAULT_BIG_WINDOW_SIZE (default 5000) Size of the sliding window in nucleotides during the calculation of the SIDD profile (**-SIDDprofile**). If the calculation takes too much time (e.g. more than a few days) a possible solution is to reduce the size of the sliding window (e.g. to 2000). It is not recommended two use a value smaller than 2000. For some organisms (e.g. S. coelicolor) a value of 10000 increased the specificity significantly while the runtime is still acceptable. However, it is advisable to adjust also the following option.

DEAFAULT_BIG_WINDOW_SHIFT (default 500) Number of nucleotides the sliding window of the SIDD profile calculation is shifted after each application. If a smaller value is chosen the calculation will take longer but is more precise. It is recommended to adjust this value in a way so that each position is covered by at least 10 windows, i.e.
DEFAULT_BIG_WINDOW_SIZE/10 or smaller.

5 GFF output

Using the **-gffOutFile** command a GFF file is created which contains all results of the applied methods. It contains predicted ncRNA regions (including classification results), protein coding regions (if provided), terminator signals, SIDD sites, sigma factor binding sites, Rfam motif matches and predicted ncRNA transcripts. For each feature type a respective example GFF

entry is shown below.

Predicted ncRNA region:

```
NC_003888.3 RNAz ncRNA_region 1 232 . . . ID "1"; Class_String "XTSX"
```

The ID of the ncRNA region is given in the attributes as well as the classification results as a string. The Class_String consists of 4 characters:
The first character indicates if the ncRNA locus overlaps a coding sequence (`G`) or not (`X`).
The second character indicates if the ncRNA locus contains a predicted terminator (`T`), a terminator that additionally indicates an antisense transcript (`A`) or no terminator (`X`).
The third character indicates if the ncRNA locus contains a predicted SIDD site (`S`), a SIDD site that additionally indicates an antisense transcript (`A`) or no SIDD site (`X`).
The fourth character indicates if the ncRNA locus contains a predicted TFBS (`P`), a TFBS that additionally indicates an antisense transcript (`A`) or no TFBS (`X`).

Protein coding region:

```
NC_003888.3 GenBank gene 421712 422263 . + . ID "SCO0400"
```

The ID of the protein coding region is given in the attributes.

Predicted terminator signal:

```
NC_003888.3 transterm terminator 1858 1870 85 + . ID "TERM6"
```

The confidence value of the terminator is given in the score field of the GFF entry. The ID is given in the attributes.

Predicted SIDD site:

```
NC_003888.3 nocoRNAc sidd_site 185398 185455 3.0 . . ID "SIDD553"
```

The incremental energy value of the SIDD site is given in the score field of the GFF entry. The ID is given in the attributes.

Detected sigma factor binding site:

```
NC_003888.3 nocoRNAc sig_binding_site 450936 450942 . + . ID "PROMCONS28";
Pattern "hrdB-35"
```

The ID and the name of the sequence pattern are given in the attributes.

Rfam motif match:

```
NC_003888.3 erpin rfam_motif 3685405 3685521 167.26 - .  ncRNA_Region "1669"
; Rfam_Seed "RF00001"; Description "5S ribosomal RNA"; E_value 5.56E-29
```

In the attributes are given:
The ID of the ncRNA region in which the motif was found the Rfam accession number of the respective seed, a brief description of the motif and the E value of the hit.

Predicted ncRNA transcript:

```
NC_003888.3 nocoRNAc ncRNA_transcript 1622761 1622950 .  + .  ID "TU443_2" ;
ncRNA_Region "443" ; SIDD_value -0.5 ; Term_Confidence 100
```

In the attributes are given:
The ID of the predicted ncRNA transcript, the ID of the ncRNA region where the transcript is located, the energy value of the SIDD site which initiates the transcript and the confidence value of the terminator which terminates the transcript. If the given confidence value equals 0, the predicted transcript has no terminator signal.

6 Integrated Programs

The external programs nocoRNAc utilizes are listed in the following. The source code has to be retrieved from the respective websites and the compiled binaries and other files needed can be placed in nocoRNAc's `progs` folder. They can also be placed somewhere else, but the respective paths have to be set in the configuration file. All binaries and Perl scripts have to be executable.

Erpin (`http://tagc.univ-mrs.fr/erpin/`)
The directory which contains the Erpin distribution has to be set in the configuration file. By default this is: `progs/erpin` The binaries have to be located in the `bin` subfolder. The Perl scripts, including `erpincommand.pl`, have to be located in the `scripts` subfolder. nocoRNAc has been tested with Erpin 5.5.

CMsearch (`http://infernal.janelia.org/`)
The path where the CMsearch binaries (`cmbuild`, `cmsearch`) can be found has also to be set in the configuration file. Default: `progs/infernal`. NOCORNAC has been tested with CMsearch 0.81.

TransTermHP (`http://transterm.cbcb.umd.edu/`)
The path where the TransTermHP binary and the needed file `expterm.dat` (contained in the TransTermHP distribution) can also be set in the configuration file. Default: `progs/transterm`. NOCORNAC has been tested with TransTermHP 2.07. Instead of using TransTermHP NOCORNAC can also import TransTermHP output files, which can be retrieved from the TransTermHP website for many bacterial genomes by the use of the **-importTransTerm** command.

IntaRNA (`http://www.bioinf.uni-freiburg.de/Software/`)
The default location of the IntaRNA binary is `progs/intarna`, but this can be changed like for the other programs. NOCORNAC has been tested with IntaRNA 1.2.1.

7 Interactive R environment

To use the interactive `R` environment you need to install `R` (`http://www.r-project.org/`; v2.8 or greater). In addition, the `Biostrings` package (Bioconductor - `http://www.bioconductor.org/`) is needed for sequence handling. In the following the most important structures and functions are described.

ncRNAs List containing information about all predicted ncRNA loci. The list entry consists of the following subentries: 'start', 'end', 'strand', 'score' (e.g. RNAz p-value), 'class' (see section 5), 'genes' (list of overlapping genes), 'terminators' (list of terminator signals associated with the locus), 'sidd.sites' (list of SIDD sites associated with the locus), 'pc.matches' (list of predicted TFBS associated with the locus), 'pred.transcripts' (list of ncRNA transcripts predicted for this locus)

nc.transcripts Table containing information about all predicted ncRNA transcripts. The rownames are the IDs of the elements. The table consists of the columns 'start', 'end', 'strand', 'sidd.site' (ID of the SIDD site), 'terminator' (ID of the terminator), 'genes' (list of overlapping genes), 'antisense' ('true' if antisense to a gene).

terminators Table containing information about all predicted terminator signals. The rownames are the IDs of the terminators. The table consists of the columns 'start', 'end', 'strand', 'score', where 'score' is the confidence value of the terminator.

sidd.sites Table containing information about all predicted SIDD sites. The rownames are the IDs of the SIDD sites. The table consists of the columns 'start', 'end', 'strand', 'score', where 'score' is the minimum free energy value of the SIDD site.

genes Table containing information about all Rfam hits. The rownames are the IDs of the hits. The table consists of the columns 'ID' (Rfam-ID), 'source.prog' (program used to match the Rfam seeds), 'nc.locus.id' (ID of the ncRNA locus), 'strand', 'start', 'end' (relative to the ncRNA locus), 'score', 'e.value' (score and E-value of the hit).

getSequences `function(features,upstream=0,downstream=0)`
This function takes sequence features from the structures described above as input (e.g. several rows from `genes`) and returns their genomic sequences as a `DNAStringSet`. The `upstream` and `downstream` parameters can be used to extract additional bases upstream or downstream of the locus.

intarna `function(hunter, target, use.window=T, full.output=F, hunter.upstream=0, hunter.downstream=0, target.upstream=0, target.downstream=0, sample.hunter=F, sample.target=F)`
This function takes the IDs of two sequence features as input (e.g. an ncRNA transcript (`hunter`) and a coding gene (`target`)) and performs an RNA-RNA interaction prediction using the program IntaRNA. The result is returned in the form of a list.